

Trabalho prático 01

UFV – Campus Florestal

Professor:	José Augusto Miranda Nacif
Disciplina:	Organização de Computadores I
Aluno:	Artur de Souza Amorim - 3048
Aluno:	Thomas Sonobe Moreira Chang - 3052
Aluno:	Victor Hugo Rezende dos Santos - 3510

Florestal, 02 de Maio de 2019



ÍNDICE

- 1. INTRODUÇÃO**
- 2. DESENVOLVIMENTO**
- 3. CONCLUSÃO**
- 4. BIBLIOGRAFIA**



1. INTRODUÇÃO

O trabalho prático foi desenvolvido com base na proposta apresentada pelo professor José M. Augusto Nacif responsável pela disciplina Organização de Computadores I (CCF252), com o objetivo de criar um montador MIPS simplificado. A proposta é a de criar o montador MIPS e converter suas entradas em suas respectivas representações em números binários.

2. DESENVOLVIMENTO

O montador traduz um arquivo de instruções em assembly para um arquivo de instruções de máquina em sua representação em binário.

O processo de tradução se consiste em receber uma instrução em uma certa linguagem e a substituir por uma outra linguagem, vale lembrar que a mensagem/instrução não se altera.

A Arquitetura MIPS codifica todas as instruções em 32 bits por convenção. Na linguagem assembly existem 3 padrões de codificação de instruções e no trabalho trataremos 2 dos 3 padrões:

- **Type-R** para instruções com operandos tipo registradores. Esse tipo de instrução segue o seguinte padrão:



Padrão de representação de instruções tipo R

rd = registrador de destino

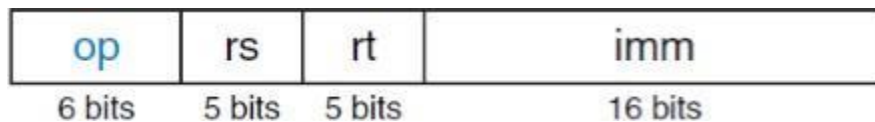
rs = primeiro registrador de origem

rt = segundo registrador de origem.

funct = determinam o tipo de operação (add, sub, etc)

shamt = shift amount é usado apenas deslocamentos

- **Type-I** para instruções com operandos imediatos. Com o seguinte padrão:



Padrão de representação de instruções tipo I

rt = registrador de destino

rs = registrador de origem

imm = constante imediata



O montador desenvolvido engloba as seguintes operações em linguagem de máquina assembly:

ADD: adição – Tipo R

SUB: subtração – Tipo R

AND: operação lógica and – Tipo R

OR: operação lógica or – Tipo R

NOR: operação lógica nor – Tipo R

SLL: deslocamento para esquerda – Tipo R

SRL: deslocamento para direita - Tipo R

ADDi: adição imediata – Tipo I

ANDi: operação lógica and imediata – Tipo I

ORi: operação lógica or imediata – Tipo I

O programa foi implementado em linguagem de programação Python (.py). As instruções em assembly devem estar em um arquivo de entrada .asm, seguindo o seu devido padrão de codificação.

```
entrada.asm x
1      add $s2, $s0, $s1
2      sll $t1, $s2, 2
3      or $t2, $s2, $s1
4      andi $t2, $t1, 16
5      addi $t3, $t2, -243
```

A aplicação lê linha por linha do arquivo .asm e traduz a instrução através da codificação apresentada no arquivo .py. Neste mesmo arquivo foram implementadas as estruturas lista e dicionário e também diferentes funções.

Após a leitura, a conversão para linguagem de máquina, binária, entra em execução. Após a conversão, a instrução binária de 32 bits, seja Tipo R ou Tipo I, é armazenada em um arquivo de saída .txt.

```
1      00000010000100011001000000100000
2      00000000000100100100100010000000
3      00000010010100010101000000100101
4      00110001001010100000000000001000
5      00100001010010111111111100001101
6      
```



3. CONCLUSÃO

O trabalho prático foi trivial para o entendimento básico do funcionamento de um montador MIPS e como são as etapas do processo de tradução de uma instrução. Ao longo do desenvolvimento do trabalho foi necessário o domínio dos tópicos abordados em sala de aula e no livro *Organização e Projetos de Computadores*, 4ªed., usado como ementa principal da disciplina.

4. BIBLIOGRAFIA

Patterson, D.A.; Hennessy J.L. **Organização e Projeto de Computadores: A interface Hardware/Software**, 4ª Edição, Editora Elsevier, 2014.

TANENBAUM, A.S. **Organização Estruturada de Computadores**. 5. ed. Editora Pearson Prentice Hall, 2007.

https://www.slideshare.net/tagbagtroj/mips-opcodes?fbclid=IwAR3Vg9avO0OGRkJnmeZEAH7y-NqwUx-JscwBsdRi-GopIDpik_sz43oGlwo. Acessado em 01 de maio de 2019.

https://en.wikibooks.org/wiki/MIPS_Assembly/MIPS_Details?fbclid=IwAR2u9ZMSK6UsVgvCvBMCEcZwsval51pvTIWqQvk8fMYx-ziVJVyPek29Eag. Acessado em 01 de maio de 2019.